

## Loops and Arrays

### Lecture 4 Assignments

**Name:** Khanne S. Labao

**Student no.:** 202209642

**Course & year:** BS in Computer Science

1. What is the output of the following program?

```
#include <stdio.h>
```

```
Int main (void)
```

```
{
```

```
    int i;
```

```
    i = 1;
```

```
    while (i <= 128) {
```

```
        print ("%d ", i);
```

```
        i *= 2;
```

```
    }
```

```
    return 0;
```

```
}
```

**Ans: 1 2 4 8 16 32 64 128**

2. Which one of the following statements is not equivalent to the other two (assuming that the loop bodies are the same)?

a.) while (i < 10) { . . . }

b.) for (; i < 10;) { . . . }

c.) do { . . . } while (i < 10);

```
1  #include <stdio.h>
2
3  int main(void) {
4
5      int i = 0;
6
7      while (i < 10) {
8          printf("%d ", i);
9          i++;
10     }
11
12     printf("\n");
13
14     i = 0;
15     for (; i < 10;) {
16         printf("%d ", i);
17         i++;
18     }
19
20     printf("\n");
21
22     i = 0;
23     do {
24         printf("%d ", i);
25         i++;
26     } while (i < 10);
27
28     printf("\n");
29
30     return 0;
31 }
```

Output:

```
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
```

When we compare the outputs of the provided loops, we cannot determine which one is different from the other two. The output of printf functions will be the same for all loops. However, we can tell them apart by how the compiler handles the loop's specific syntaxes. The do-while loop checks the stopping condition at the end of the iteration, whereas the other two loops check it at the start.

3. Convert item 1 into an equivalent for statement. You can validate your answer by checking if the produced outputs by both the while and for statements are similar.

Ans:

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int i;
6      i = 1;
7      for (i=1; i<=128; i*=2) {
8          printf("%d ", i);
9      }
10
11     return 0;
12 }
13
```

Output:

```
1 2 4 8 16 32 64 128
```

4. Write a code that computes for the power of two.

Ans:

```
1  #include<stdio.h>
2
3  int main() {
4      long int p;
5      int n;
6
7      printf("Table of Powers of Two\n\n");
8      printf("\n n    2 to the n");
9      printf("\n== =====");
10
11     p = 1;
12     for (n = 0; n < 11; ++n) {
13         if (n == 0)
14             p = 1;
15         else
16             p = p * 2;
17         printf("\n%2d    %8ld", n, p);
18     }
19     return 0;
20 }
21
```

Output:

Table of Powers of Two

n	2 to the n
=====	
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

5. Write a program that displays a one-month calendar.

Ans:

```

1  #include <stdio.h>
2
3  int main(void)
4  {
5      int days, start_day, i, j;
6
7      printf("\nEnter number of days in month: ");
8      scanf("%d", &days);
9
10     if (days != 28 && days != 29 && days != 30 && days != 31)
11         printf("Invalid number. Please try again.");
12
13     else
14     {
15
16         printf("Enter starting day of the week (1=Sun, 7=Sat): ");
17         scanf("%d", &start_day);
18         printf("\n");
19
20         if (start_day <= 0 || start_day >= 8)
21             printf("Invalid number. Please try again.");
22
23         else
24         {
25             /* prints the blank days of the first week */
26             for (i = 1; i < start_day; i++)
27                 printf(" ");
28
29             /* prints the calendar numbers */
30             for (j = 1; j <= days; i++, j++) {
31                 printf("%3d", j);
32                 if (i % 7 == 0)
33                     printf("\n");
34             }
35
36             printf("\n\n");
37         }
38     }
39
40     return 0;
41 }

```

Output:

```

Enter number of days in month: 31
Enter starting day of the week (1=Sun, 7=Sat): 3

    1  2  3  4  5
  6  7  8  9 10 11 12
 13 14 15 16 17 18 19
 20 21 22 23 24 25 26
 27 28 29 30 31

```

6.

- a. Revise line 16 such that you use a designated initializer to set pathways 0 and 2 to true, and the rest will be false. Make the initializer as short as possible.

```

1  #include <stdio.h>
2  #include <stdbool.h>
3
4  #define NUM_PATHWAYS ((int) (sizeof(pathway) / sizeof(pathway[0])))
5
6  int main(){
7      bool pathway [8] = {[0]= true,[2] = true};
8
9      for (int i = 0; i < NUM_PATHWAYS; i++){
10
11         if (pathway[i]){
12             printf("pathway [%d] is open \n", i);
13         }else{
14             printf("pathway [%d] is close \n", i);
15         }
16     }
17     return 0;
18 }

```

Ans: `bool pathway [8] = {[0]= true,[2] = true}`

- b. Revise line 16 such that the initializer will be short as possible (without using a designated initializer)

```
1  #include <stdio.h>
2  #include <stdbool.h>
3
4  #define NUM_PATHWAYS ((int) (sizeof(pathway) / sizeof(pathway[0])))
5
6  int main(){
7      bool pathway [8] = {true, false, true};
8
9      for (int i = 0; i < NUM_PATHWAYS; i++){
10         if (pathway[i]){
11             printf("pathway [%d] is open \n", i);
12         }else{
13             printf("pathway [%d] is close \n", i);
14         }
15     }
16
17     return 0;
18 }
19
```

Ans: **bool pathway [8] = {true, false, true};**

7.

As a programming assignment:

1. Declare and initialize a road\_networks multidimensional array that represents the adjacency matrix
2. Display the adjacency matrix. Put a bracket to the points/destinations that are considered as charging stations, e.g. [c], [d]
3. Given a point / destination, determine the nearest charging station. For example, if you are in point a, the nearest charging station is point c. If you are in point e, the nearest charging station is point d.
4. Bonus: Use a macro to define the size of the 2d array.



